SYSTEM AND METHOD OF ANALYZING AN HTML DOCUMENT FOR CHANGES SUCH THAT THE CHANGED AREAS CAN BE DISPLAYED WITH THE ORIGINAL FORMATTING INTACT

Background of the Invention

This invention relates generally to a system and method for comparing the differences between two documents and in particular to a system and method for comparing two hypertext markup language (HTML) documents and displaying the changed areas in the HTML documents while retaining the original HTML formatting.

The traditional method of locating document changes within pure text files is accomplished via a technique known as file differencing, or diffing. UNIX has a utility called "diff" that is used for file differencing. It works by comparing each line in a first file (the Right File) with each line in a second file (the Left File). A carriage return character typically separates each line from each other line. After the comparisons are finished, each line in the Right File will be identified as having one of the following states:

1. unmodified – The current line exactly matches another line in the Left File;

2. new – The current line has no match in the Left File; and

3. modified – The current line nearly matches one of the lines in the Left File with some changes.

The unit of comparison, a line, is deliberately chosen because it is an intermediate amount of information. In other words, it is somewhat larger than a single character or word, and therefore offers a meaningful context for the detected change. However, a line is still small

enough so that the remainder of text, divided into lines, is considered separately and most lines are often identical in both files.

To better understand this typical differencing technique, consider a "diff" operation, i.e., line-by-line comparison, of the text on the left and its revised version on the right where modified lines have been underlined in the right file for emphasis:

La Nina continues in the Pacific Ocean, meaning cooler than average sea surface temperatures along the equator north of South America. Typically this means a warmer and drier summer for the Midwest. The Summer of '99 has been very hot, with 32 days recording highs of 90 degrees or above, and very dry, with rainfall deficits exceeding 4.5 feet so far.	La Nina continues in the Pacific Ocean, meaning cooler than average sea surface temperatures along the equator west of South America. Typically this means a warmer and drier summer for the Midwest. The Summer of 1999 has been very hot, with 32 days recording highs of 90 degrees or above, and very dry, with rainfall deficits exceeding 4.5 inches so far.

For this example, Table 1 below illustrates what you would see if the basis of comparison was a word (left column) vs. a line (right column).

Table 1 – Changes Found After Comparing Text on a Word Basis vs. Line Basis

Word Basis	*Line Basis*
west	Equator west of South America
1999	Summer of 1999 has been very hot
inches	Rainfall deficits exceeding 4.5 inches

As illustrated by the above example, the detected changes in a line by line based comparison (right column) are more useful for conveying the essence of the revisions than the detected changes when using a smaller unit comparison, such as a word based comparison.

The World Wide Web (Web) is an international network of computers containing a vast amount of information. The hypertext markup language (HTML) is the lingua franca for publishing documents on the Web. The problem is that the typical differencing operations as described above do not work well for HTML documents. In particular, unlike pure text documents, or documents created using a word processor, carriage returns in HTML documents are not significant. In more detail, the width of lines displayed by a viewer will be determined by the width of the viewer window, not where carriage returns are entered in the HTML file. Therefore, a typical differencing operation that uses lines for a unit of comparison does not work successfully when comparing HTML files since the operation may unnecessarily identify differences which are insignificant. In addition, the HTML language treats contiguous sequences of white space characters as being equivalent to a single space character. Therefore, a contiguous sequence of white space characters is equivalent to a single white space character in the HTML language, but a typical differencing operation will not take this into account.

Due to the peculiar rules of the HTML language described above, the following are equivalent representations of the same paragraph in HTML document sources:

Example 1a – HTML Paragraph

<P> La Nina continues in the Pacific Ocean, meaning cooler than average sea surface temperatures along the equator west of South America. Typically this means a warmer and drier summer for the Midwest.
<\P>

Example 1b – Equivalent Variation of HTML Paragraph

<P> La Nina continues in the Pacific
Ocean,
meaning cooler than
average sea surface temperatures along
the equator west of South America. Typically this means a warmer and drier summer for the Midwest.
</P>

If a typical differencing operation is carried out on the two above paragraphs (which are considered to be identical in the HTML language), the differencing operation would find multiple differences since each line is compared character by character. It is thus apparent that applying a typical differencing operation to these HTML formatted paragraphs would be ineffective, as it would identify every line as changed instead of recognizing these representations as equivalent. Thus, it is desirable to provide a system and method for analyzing an HTML document for changes and for displaying the changed areas with the original HTML formatting intact and it is to this end that the present invention is directed.

Summary of the Invention

The World Wide Web (Web) is an international network of computers containing a vast amount of information. HTML is the lingua franca for publishing documents on the Web. This invention describes a computer-based method for analyzing two versions of an HTML document, which identifies new or changed areas of the document while preserving the original textual formatting.

Using the system and method in accordance with the invention, two versions of an HTML document are to be analyzed. The original version will be referred to as the Left File, while the updated version will be referred to as the Right File. Possible modifications of the Left File to produce the Right File might include the deletion of text, hypertext links, or embedded images; the modification of text, hypertext links, or embedded images; the insertion of text, hypertext links, or embedded images; or any combination thereof. These document elements are usually the most interesting elements for users to monitor for changes, but any document element can be monitored for changes with this method, while preserving visual formatting in the vicinity of the change or changes. Examples of visual formatting include font type, font size, and use of bold or italics.

In more detail, an HTML document may be scanned and the information organized into groupings of HTML tags and text. The system includes a set of rules for determining which HTML tags are permitted within a group, and which HTML tags mark the start of a new group. The tags that mark the start of a new group are usually those that break the flow of text when an HTML page is rendered. As a result, the text that constitutes a paragraph, embedded hypertext links, and any associated HTML character-formatting elements are contained within a single group. A modified version of the same HTML document is similarly processed. Once the processing is complete, the two HTML documents may be compared group by group in order to detect differences. Any group that does not match the associated group in the original is considered to be a modified group. The modified groups can then be inserted as sections into a new HTML document, and these sections appear to have nearly all of the original formatting

intact. Thus, the modified sections may appear as clipped sections from the original HTML document and are useful for depicting regions of interest. In addition to providing a new document with the clipped sections, a HTML page with the changed highlighted for the user may be displayed to the user.

Brief Description of the Drawings

Figure 1 is a diagram illustrating an example of a computer-based system that may be used to execute the HTML normalization method in accordance with the invention;

Figure 2 is a diagram illustrating a computer-implemented HTML normalization and comparison system in accordance with the invention;

Figure 3 is a flowchart illustrating a method for HTML normalization in accordance with the invention;

Figure 4 illustrates an example of a left file in accordance with the invention;

Figure 5 illustrates an example of a right file in accordance with the invention;

Figure 6 illustrates an example of a typical HTML file;

Figure 7 illustrates the HTML file of Figure 6 after normalization in accordance with the invention; and

Figure 8 illustrates an example of the comparison results file being displayed to the user.

Detailed Description of a Preferred Embodiment

The invention is particularly applicable to a personal computer based system and method for normalizing and comparing HTML documents and it is in this context that the invention will be described. It will be appreciated, however, that the system and method in accordance with the invention has greater utility, such that it may be implemented using other types of computer systems, such as a client/server type system or any other computer-based system and may be used with other formatted files.

Figure 1 is a diagram illustrating an example of a computer-based system 10 that may be used to execute the HTML normalization method in accordance with the invention. In this example, a typical personal computer is shown although the system and method in accordance with the invention may also be implemented on other different types of computer systems, such as client/server systems, local area networks and the like. The computer 10 may include a display unit 12, a main processing unit 14 and one or more input/output devices 16. In this example, the one or more input/output devices may include a keyboard 18 and a mouse 20. In accordance with the invention, the input/output devices may also include, for example, a printer. The display unit 12 may be any typical display device, such as a cathode ray tube, a liquid crystal display or the like.

The main processing unit 14 may further include a central processing unit (CPU) 22, a memory 24 and a persistent storage device 26 that are interconnected together. The CPU 22 may control the operation of the computer and may execute one or more software applications, such as the HTML normalizer and comparer in accordance with the invention. The software

applications may be stored permanently in the persistent storage device 26 that stores the software applications even when the power is off and then loaded into the memory 24 when the CPU is going to execute the particular software application. The persistent storage device 26 may be a hard disk drive, an optical drive, a tape drive or the like. The memory may be a random access memory (RAM), a read only memory (ROM) or the like. In operation, a normalization and comparing software application may be stored in the persistent storage device and, based on user input, loaded into the memory to be executed by the CPU. The normalizer and comparer system in accordance with the invention may normalize the HTML documents, as described below, into one or more blocks of information and then compare the blocks of information to each other in order to accurately compare the HTML documents and maintain the formatting of the HTML documents during the comparison. Now, more details of the normalization and comparison system in accordance with the invention will be described.

Figure 2 is a diagram illustrating a computer-implemented HTML normalization and comparison system 30 in accordance with the invention. Although a software application implemented system and method in accordance with the invention is described herein, the system and method may also be implemented in hardware. The system 30 may include one or more software application modules that may be executed by the CPU (See Figure 1) in order to perform the functions of the system in accordance with the invention. The system 30 may include an HTML normalizer module 32, a rules database 34 and a comparer 36. The normalizer module 32 may convert a first HTML document (HTML #1) and a second HTML document (HTML #2) into a normalized first and second document (RIGHT and LEFT) based on one or

more normalization rules that may be stored in the rules database 34. The normalization of the two HTML documents permits those documents to be compared by a typical line comparison module 36 and then to display the results of the comparison while maintaining the formatting on the HTML documents. In general, the normalization may involve the conversion of the HTML document into one or more blocks of information wherein each block of information may be treated as a single line for purposes of the comparison. Thus, the normalization permits a typical line based comparison module to be used to accurately compare two HTML documents. More details of the normalization and the normalization rules in the rules database will now be described.

Figure 3 is a flowchart illustrating a method 40 for HTML document normalization in accordance with the invention so that two HTML documents may be compared to each other using typical comparison systems while maintaining the formatting of the HTML documents. In step 42, the entire HTML document is scanned and any HTML head element are removed from the document. In step 44, the HTML document is scanned again and any references to scripts in the HTML document are removed. In step 46, the HTML document is scanned again and any intradocument links are removed. In step 48, the HTML document is scanned again and any relative URLs in the HTML document are converted into absolute URLs. These rules in steps 42 – 48 provide special handling of HTML elements which are not valid when removed from the context of the original page.

Now, the entire HTML document is scanned again on a character by character basis to complete the normalization process. In particular, in step 50, the next character in the document

is retrieved. Next, in step 52, the method determines if a preformatted text character sequence (/PRE) has been located by scanning several characters. If the preformatted text tag has not been located, then character by character processing in step 54 occurs. The character by character processing will be described below in more detail. If a preformatted text tag has been located, then the method skips step 54 so that none of the character by character processing is carried out on the preformatted text. With the test in step 52, once an end tag for the preformatted text is located, the character by character processing may be resumed. Next, in step 56, the method determines if there are more characters to analyze and loops back to step 50 to get the next character or the normalization process is completed.

The character by character processing may occur by applying multiple different rules to each character. The rules may include removing the carriage returns from the HTML document, converting multiple white spaces in the document into single white spaces, separating any block level HTML elements from each other onto separate lines by inserting a carriage return before the start tag so that each block in the HTML document is treated as a separate line for comparison purposes, and keeping any text level HTML elements on the same line. It should be noted that text level HTML elements don't cause paragraph breaks when rendered into a displayable form in a web browser. Those text level elements that define character styles can generally be nested as long as they contain other text level elements but not block level elements since block level elements are placed on separate lines. In accordance with the invention, the text level elements may include, for example, font style elements, phrase elements, form fields, A (anchor) elements, IMG elements (e.g., an inline image in an HTML document), APPLET

elements (e.g., Java Applets), FONT elements, BASEFONT elements, BR elements (e.g., line breaks in the HTML document), and MAP elements (e.g., a client-side image map in the HTML document). In removing the white spaces, the method may encounter a first white space and store it and then throw away all subsequent white spaces until another character is encountered so that the multiple contiguous white space characters are converted into a single white space character.

The normalization process has now been completed so that the two normalized HTML documents may be compared by a typical line comparison operation while still maintaining the formatting of the HTML document. In summary, the invention establishes a method for generating a normalized form for an HTML document so that equivalent representations, once normalized, will appear identical when analyzed via typical line differencing. Another result is that normalization, when applied as described, will organize block elements, which do not nest additional block elements, each on separate lines. This is important because it keeps hypertext links and presentation elements, those that produce visual effects, together with textual content. Thus, when line differencing is applied, the detected changes are at the block level. In addition. these block elements will be properly formed HTML, with textual formatting automatically included along with any associated text.

As described above, there are additional rules that make reference to other areas within the HTML document, such as removing intradocument links and script references and converting relative URLs. These rules are not strictly necessary to facilitate the comparison process, but do avoid errors in the rendering stage when the changed blocks of HTML, inserted into the body of

a new HTML document, are rendered in the browser. The head element is also removed since this area of the HTML source does not contain displayable information, and so it isn't usually useful to report changes occurring here.

Now, an example of two HTML documents being normalized in accordance with the invention will be described. Referring back to Examples 1a and 1b above, if the rules of normalization in accordance with the invention are applied to Examples 1a and 1b, the two representations would appear identical. In particular, for this example, all characters would be organized on a single line, including the <P> and </P> tags.

In accordance with the invention, the changed areas that are discovered through the differencing process appear as clipped regions from the original revised HTML document (Right File) when rendered in the browser. Instead of the clipped regions, a new HTML page with the all of the original content plus the changes highlighted may also be displayed for the user. Now, an example of the system and method in accordance with the invention will be described.

Figure 4 illustrates an example of a left file 60 and Figure 5 illustrates an example of a right file 62 wherein the two files are HTML pages that display information to the user. For purposes of a simple example, there is additional text in the second paragraph of the right file shown in Figure 5, beginning with the sentence formatted in bold. The second paragraph is the changed information that will be automatically identified by the system and method, along with its embedded formatting, in accordance with the invention. The system and method can obviously also be used to compare more complex HTML files.

Figure 6 illustrates an example of a typical HTML file 64 that represents the left file shown in Figure 4 while Figure 7 illustrates a normalized HTML file 66 that corresponds to the HTML file shown in Figure 6. As described above, various elements of the HTML file are removed, not to facilitate the comparison process, but to avoid errors in the rendering stage when the changed blocks of HTML are inserted into the body of a new HTML document for display in the browser. As also explained, each paragraph shown in Figures 4 and 5 is inside a block-level HTML element that is deliberately arranged on a single line. Thus, although various elements of the HTML page are removed, portions of the HTML are arranged on a single line so that the line-by-line comparison method may be used to find differences in the HTML documents.

Figure 8 illustrates an example of the comparison results file 80 being displayed to the user. As shown, the file may be a HTML page that is displayed to the user. As shown, the formatting of the changed portion (part of which has a bold appearance) and the surrounding portions is maintained so that the user can see the new or changed information as it was originally intended to appear.

While the foregoing has been with reference to a particular embodiment of the invention, it will be appreciated by those skilled in the art that changes in this embodiment may be made without departing from the principles and spirit of the invention, the scope of which is defined by the appended claims.